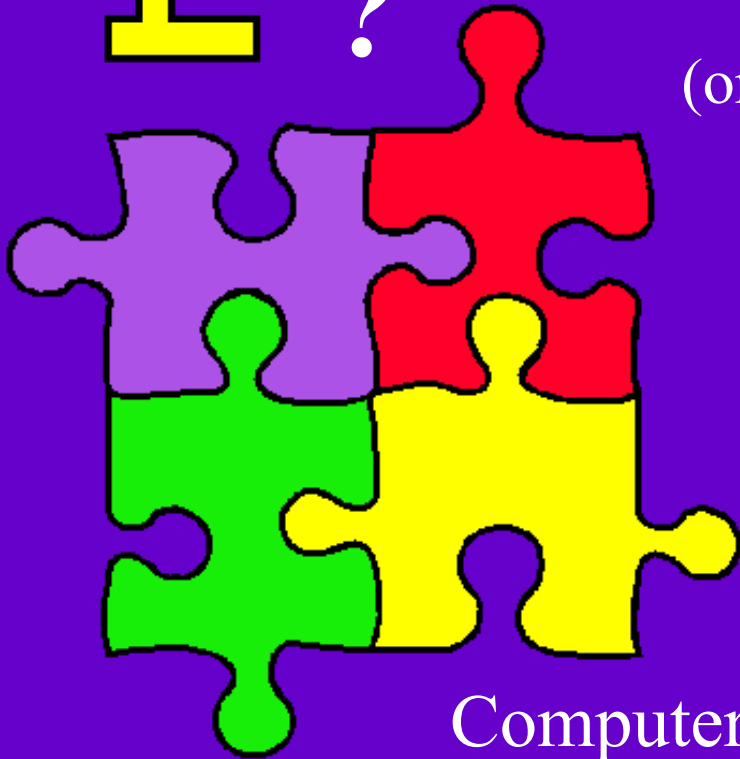


?

# Development of Compatible Component Software

(or CUMULVS, CCA, Harness, PVM, MPI  
and other assorted acronyms... ☺)



Tuesday, July 24, 2001

James “Jeeembo” Kohl

Computer Science and Mathematics Division

Oak Ridge National Laboratory

# What a Great Idea!



- Let's Componentize Software!
- Experts Write Many Specialized Components
  - ⇒ Pick the Right Component, Then Plug & Chug...
- Software Re-Use, Interoperability
  - ⇒ Define “Ports” for Component Interfaces
  - ⇒ Connect Up Ports to Assemble Applications
- What Can Possibly Go Wrong?
  - ⇒ Go Wrong... Go Wrong... Go Wrong...
  - (echoes from 20 years ago... still resonating... ☺)

# Reality of Software Development

- Vendors Customize Operating Systems
  - ⇒ Every Hardware Architecture is Different
- Version 1.1 May Not Be Compatible with 1.0
  - ⇒ Software Developers Don't Always Look “Back”
  - ⇒ True of Both O.S. and Tools...
- My Dad Can Beat Up Your Dad!
  - ⇒ Always Plenty of “Standards” to Choose From
  - ⇒ Caught Wearing the Same Costume to the Ball
    - Force Interoperability After the Fact... huhuhuhuh...

# “War” Story: PVM and MPI



- PVM ~ Research Project, Started ‘89:
  - ⇒ Message-Passing, Task & Resource Management, Heterogeneity, MPPs, Fault-Tolerance...
- MPI ~ Message-Passing “Standard”, ‘93:
  - ⇒ Comprehensive Point-to-Point, Collective, Context, Communicators, Topologies, Profiling
- MPI\_Connect / “PVMPI”, ’95...
- MPI-2 ~ When 250 Routines Isn’t Enough, ’95:
  - ⇒ One-Sided, Spawn, I/O, Language Interop...
- IMPI ~ Interoperability Among MPIs, ’97...

# PVM versus MPI Reality...

- Different Design Goals:
  - ⇒ Research Platform versus Ultimate Performance
    - PVM ~ Dynamics and Flexibility, Fault Recovery...
    - MPI ~ Fast, Static Model; No Daemons... (yet)
- Different Applications Have Different Needs
  - ⇒ Either PVM or MPI could be the “Right Choice”
- Some “Convergence” Over Time...
  - ⇒ PVM has added Context...
  - ⇒ MPI has added Spawn...
- Coexistence...

# PVM versus MPI ~ Ramifications

- There is No ONE Message-Passing “Standard” that Pleases Everyone...
- Distributed Computing Tools Must Therefore:
  - ⇒ Support BOTH Systems, Or...
  - ⇒ Choose ONE and Abandon Users of the Other...
- Is This So Terrible?
  - ⇒ Users “Mostly” Satisfied, One Way or Another...
  - ⇒ Compatibility and Interoperability? Um, NOPE.

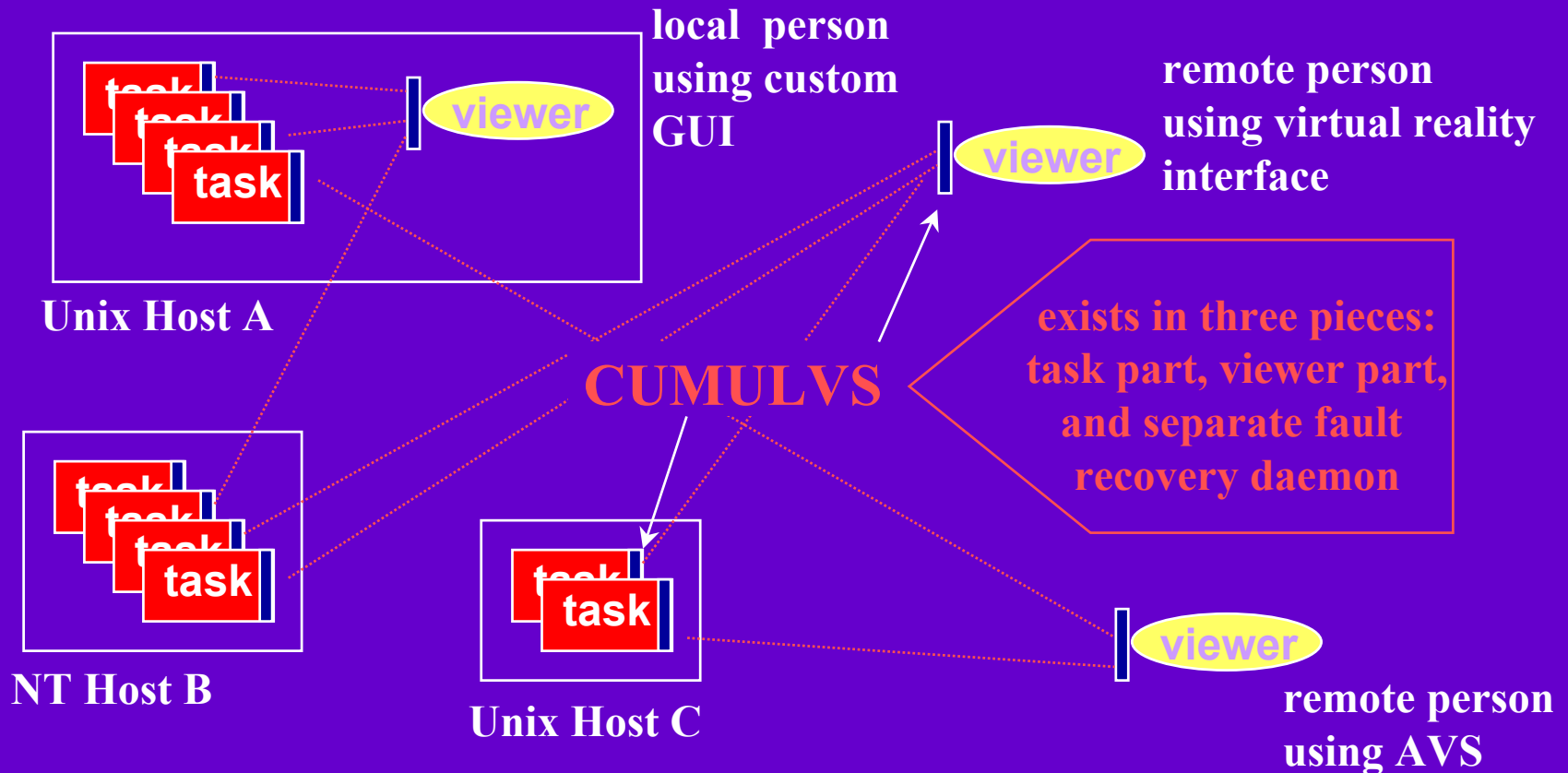


(Collaborative, User Migration, User Library for Visualization and Steering)

- Collaborative Infrastructure for Interacting with Scientific Simulations On-The-Fly:
  - ⇒ Run-Time Visualization by Multiple Viewers
    - Dynamic Attachment
  - ⇒ Coordinated Computational Steering
    - Model & Algorithm
  - ⇒ Heterogeneous Checkpointing / Fault Tolerance
    - Automatic Fault Recovery and Task Migration
  - ⇒ Coupled Models...

# CUMULVS

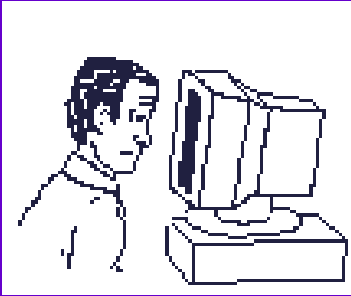
**coordinates the consistent collection and dissemination of information to / from parallel tasks to multiple viewers**



distributed parallel application or simulation

supports most target platforms (PVM / MPI, Unix / NT, etc.)





# CUMULVS “War” Stories

(of which there are many... ☺)

- CUMULVS and MPI
  - ⇒ Nexus / Mpich-G
  - ⇒ MPI-1
  - ⇒ MPI-2
- CUMULVS and Global Arrays
- CUMULVS and CCA

# CUMULVS & MPI

- CUMULVS Originally Written Over PVM
  - ⇒ Dynamic Attachment ~ Independent Spawns
  - ⇒ Application Discovery ~ Key-Value / Mbox
  - ⇒ Fault Tolerance / Recovery ~ Notify, Host Add
  - ⇒ Works Great! (for applications written in PVM )
- What About MPI Applications? Hmmmm...
  - ⇒ MPI Tasks Also Register as PVM Tasks
  - ⇒ Application Uses MPI / CUMULVS Uses PVM
  - ⇒ Yechhhh... Functional, But Not User Friendly

# Solution #1: CUMULVS & Nexus

- MPI Doesn't Support:
  - ⇒ Independent Spawn, Application Discovery, Fault Tolerance (with some upcoming exceptions...)
- Mpich-G (Globus) Built on Nexus
  - ⇒ Nexus Supports Some Dynamics and Fault Notify
  - ⇒ Port CUMULVS to Nexus
  - ⇒ Application and “Viewer” Use Mpich-G! ☺
- What a Great Idea! (Uh-Oh...)





# There's Some Bad News, and Some Worse News...

- Nexus is Message-Handler Based...
  - ⇒ COMPLETELY! No Point-to-Point Messaging!
  - ⇒ O.K., so Convert CUMULVS to ALL Message-Handler Based Messaging...
    - Not so Bad, Good for Fault Tolerant State Protocols
    - PVM 3.4 Supports a Form of Message Handlers, too...
    - Lotsa Work, But Well Justified. We Did It.
- BUTT, Nexus is DEAD?! Bummer... :-o
  - ⇒ Globus Team Decided to Replace Nexus
  - ⇒ Simple Messaging System in its Place... D-Oh!

# Back to Square “One” (MPI-1)

- O.K., What If CUMULVS Wasn’t So “Cool”? ☺
  - ⇒ Can We Still “Attach” to an MPI-1 Application?
- MPI-1 Has No “Spawn”...
  - ⇒ Something Has To Be There All Along...
    - Start Extra CUMULVS Proxy Task in MPIRUN?
  - ⇒ What About MPI\_COMM\_WORLD?
    - How to Exclude the Proxy Task from Collective Calls?
    - Force Applications to Use Alt World Communicator?!
- Requires Good Ole TCP/IP for Viewer Conns
  - ⇒ Bogus, But Doable... (work in progress)

# Square “Two” (MPI-2)

- MPI-2 Has SPAWN! :-D
  - ⇒ Now We Can “Attach” Dynamically!
  - ⇒ What a Great Idea! (Uh-oh... ☺) 
- Hmmm... Just Like “Get Smart” Cyanide Pill:
  - ⇒ “How Do We Get Them to Take It?” ☺
  - ⇒ Need Some Way to Externally “Trigger” Collective Spawn in MPI-2 Application... 
  - ⇒ File-Based Flag? Another Simple TCP/IP Hook?
  - ⇒ Good “Student” Project... Heh, heh, heh...

# Another Tool Interoperability Study: CUMULVS & Global Arrays

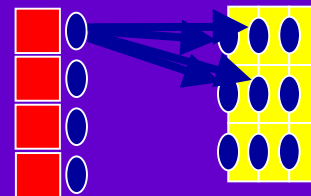
- Proof of Concept ~ Jarek @ PNNL
- Production Interface ~ David @ ORNL
- Idea: Wrap Global Arrays for CUMULVS
  - ⇒ Eases CUMULVS Instrumentation Hassle
  - ⇒ CUMULVS Library Extensions for GA
    - “One-Call” Field Definitions for CUMULVS
- SUCCESS! This Idea “Just Worked”... 😊
  - ⇒ Minor Glitches: Row Major / Column Major, Processor Topology Munging...

# CUMULVS & CCA

- Symbiotic Relationship:
  - ⇒ CUMULVS Drives “MxN” Development...
  - ⇒ CCA MxN Interface “Stretches” CUMULVS
  - ⇒ (Merging with Other Tools ~ PAWS, Meta-Chaos)
- SC Demos ~ CUMULVS “Eyes” Component
  - ⇒ SC99: Wrapped CUMULVS as One Component
  - ⇒ SC00: Started Pulling Apart CUMULVS
    - “Data Holders” Interfaced to CUMULVS (Like GA)
    - New “Eyes” Component Just Handled Comm
  - ⇒ SC01: First Shot at “Real” MxN Interface...



# CUMULVS & MxN



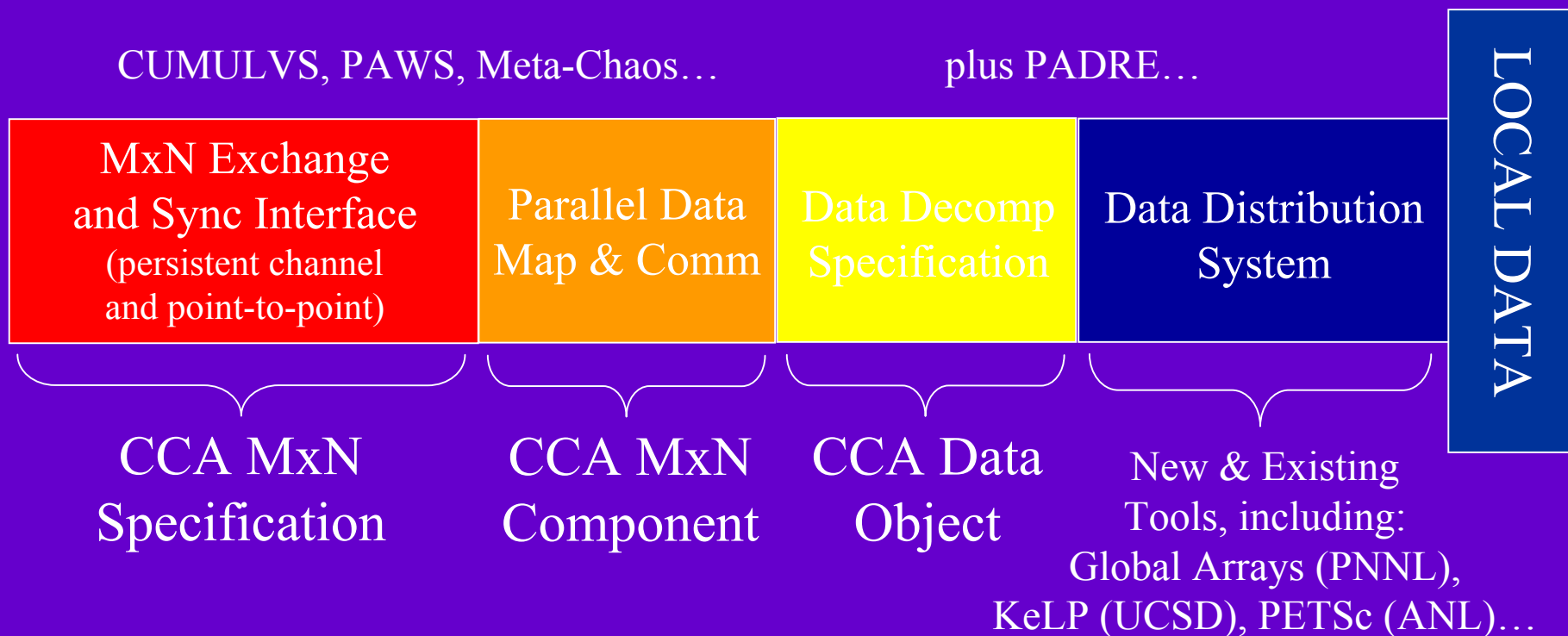
- MxN Parallel Data Redistribution Interface
  - ⇒ Specification “0.5” Hot Off the Press... ☺
- Integrates CUMULVS, PAWS, Meta-Chaos
  - ⇒ Several Approaches/Variations ~ One Interface
    - “One-Shot” Transfers, No Persistent Synchronization
    - “Periodic” Transfers, Ongoing Data Frames, a la Viz
  - ⇒ Solidify & Formalize Internals ~ Comm Sched
- No One Tool Fully Implements MxN!
  - ⇒ Everyone has to Stretch Their Tools...
  - ⇒ Multi-Stage Development & Deployment

# MxN Technology Sources

CUMULVS (ORNL):

MxN Exchange and Sync Interface (persistent channel)	Parallel Data Map & Comm	Data Decomp Specification	Global Arrays (PNNL)  PVM	LOCAL DATA
PAWS (LANL):			MPI	
MxN Exchange and Sync Interface (point-to-point)	Parallel Data Map & Comm	Data Decomp Specification	KeLP (UCSD)	
PADRE (LANL):			PETSc (ANL)	
	Parallel Data Map & Comm	Data Decomp Specification	Data Distribution System	
Meta-Chaos (U Maryland):				
MxN Exchange and Sync Interface (point-to-point)	Parallel Data Map & Comm	Data Decomp Specification	Other Data Dist Tools...	

# MxN Technology Integration



# CUMULVS & CCA Data

- CCA Generalized Data Interface
  - ⇒ Interface for Structured / Unstructured Meshes
  - ⇒ Use to Describe Existing / Develop New Data
- Existing “MxN” Tools:
  - ⇒ Will Require Modification for New Mesh Types
    - Nobody (in MxN) Handles Unstructured Very Well Yet
      - \* CUMULVS “Particle” Interface...
  - ⇒ Benefit From “Standardized” Interface
    - Now We Have a “Chance” at Handling Unstructured!
- CCA Data Should Ease Data Instrumentation...

# Harness\*: Parallel Plug-in Virtual Machine Research

Building on our experience and success with PVM...

- Create a fundamentally new heterogeneous virtual machine
- Based on three research concepts:

- **Parallel Plug-in Environment**

Extend the concept of a plug-in to the parallel computing world.

- **Distributed Peer-to-Peer Control**

No single point of failure unlike typical client/server models.

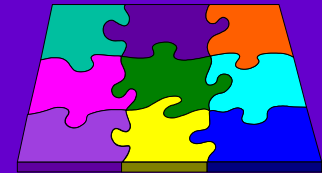
- **Multiple Distributed Virtual Machines Merge/Split**

Provide a means for short-term sharing of resources and collaboration between teams.

\* Collaborative Project ~ ORNL, UTK and Emory.

# Harness Motivated By Needs From Simulation Science

- Develop applications by plugging together ***Component Models!***




- Customize/tune virtual environment for application's needs and for performance on existing resources.
- Support long-running simulations despite maintenance, faults, and migration (dynamically evolving VM).
- Adapt virtual machine to faults and dynamic scheduling in large clusters (DASE).

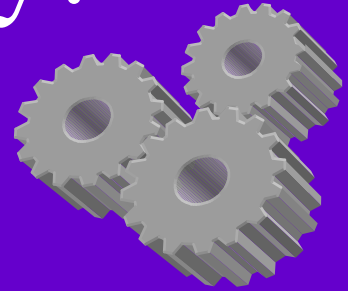


- Provide framework for collaborative simulations (in spirit of CUMULVS).

# Subtle Warning: Harness

- Harness Concept is Simple, Yet Powerful
  - ⇒ Harness is Effectively a Component Framework
  - ⇒ (What a Great Idea! ☺) 
- BUTT, the Devil is in the Details:
  - ⇒ Huge Number of “Base” Components Needed
  - ⇒ The “Noose” of Flexibility Hangs High...
  - ⇒ Dynamic Pluggability
    - Can \*Everything\* be a “Plug-In”? System Modules?
  - ⇒ How Much Can Be Made Truly “Reusable”?

# True Component Compatibility?



- Will Component *A* Ever Be Re-Used?
  - ⇒ Designed for Use with Component *B*...
  - ⇒ Component *C* has Something Different in Mind...
- True Re-Use Requires “Standards”:
  - ⇒ Specific Method Invocation Interfaces
  - ⇒ Data Description Interfaces
  - ⇒ Semantic Standards? Yikes! (Run Away! ☺)
- Each Application Domain Must Have Them!
  - ⇒ If We’re Lucky, Some General Interfaces Will Propagate... Go SciDAC Go! :-D



# Summary



- Obstacles to Overcome  
in “Componentland”
  - ⇒ Multiple “Overlapping” Solutions Already Exist
    - PVM & MPI... CUMULVS & PAWS...
      - \* Sometimes Co-Exist... Sometimes Merge Functionality...
  - ⇒ Not All Software Systems are “Compatible”
    - CUMULVS & MPI... Specific Components...
      - \* Need Application Domain-Specific Interface “Standards”...
  - ⇒ Nothing Lasts Forever ~ Platform Selection...
    - “Nexus is Dead”... C++ vs. Java vs. Python...?
- CCA ~ On Right Track; Work Cut Out For Us!

# Unanswered Killer Question...

- How Can Rob Still Look So Young & Sexy When He's Been In The Game For So Long?